ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

Frequent Item Mining Using Damped Window Model

K.Kannika Parameswari[1], Dr.Antony Selvadoss Thanamani[2]

Research Scholar, Dr.Mahalingam Centre for Research and Development, NGM College, Pollachi, India[1]

Associate Professor, Dr. Mahalingam Centre for Research and Development, NGM College, Pollachi, India[2]

Abstract: In the modern days, streams of data can be constantly generated by sensors in various real-life applications such as environment surveillance. Due to the continous flow of transactions, data in these streams can be uncertain. To discover useful and potential knowledge in the form of frequent patterns from streams of uncertain data, a few algorithms have been developed. Most of the algorithm use the sliding window model for processing and mining data streams. However, for some applications, other stream processing models such as the time-fading model are more appropriate. In this paper, we propose mining algorithms that use the damped model to discover frequent patterns from streams of uncertain data.

Keywords: Knowledge discovery, data mining techniques, data streams, frequent item sets, probabilistic data

I. INTRODUCTION

Frequent pattern mining helps discovers implicit, previously unknown, and potentially useful knowledge in the form of frequently occurring sets of items that are embedded in the data. For example, it finds from shopping market basket data those sets of popular merchandise items, which in turn helps reveal shopper behaviour. Nowadays, the automation of measurements and data collection is producing tremendously huge volumes of data. For instance, the development and increasing use of a large number of sensors (e.g., electromagnetic, mechanical, and thermal sensors) for various real-life applications (e.g., environment surveillance, manufacture systems) have led to *data stream*. To discover useful knowledge from these streaming data, several mining algorithms have been proposed. In general, mining frequent patterns from dynamic data streams is more challenging than mining from traditional static transaction databases due to the following characteristics of data streams:

1. *Data streams are continuous and unbounded*. As such, we no longer have the luxury to scan the streams multiple times. Once the streams flow through, we lose them. We need some techniques to capture important contents of the streams. For instance, *sliding windows* capture the contents of a fixed number (*w*) of batches (i.e., *w* most recent batches) in the streams. Alternatively, *landmark windows* capture contents of all batches after the landmark

Frequent Pattern Mining from Time-Fading Streams of Uncertain Data (i.e., sizes of windows keep increasing with the number of batches). Similarly, *time-fading windows* also capture contents of all the batches but weight recent data heavier than older data (i.e., monotonically decreasing weights from recent to older data).

2. *Data in the streams are not necessarily uniformly distributed.* As such, a currently infrequent pattern may become frequent in the future and vice versa. We have to be careful not to prune infrequent patterns too early; otherwise, we may not be able to get complete information such as frequencies of some patterns (as it is impossible to recall those pruned patterns).

II. LITERATURE REVIEW

J.H. Chang and W.S. Lee Knowledge embedded in a data stream is likely to be changed as time goes by. Identifying the recent change of the knowledge quickly can provide valuable information for the analysis of the data stream. However, most mining algorithms over a data stream are not able to extract the recent change of knowledge in a data stream adaptively. This is because the obsolete information of old data elements which may be no longer useful or possibly invalid at present is regarded as being as important as that of recent data elements. This paper proposes a sliding window method that finds recently frequent item sets over a transactional online data stream adaptively. The size of a sliding window defines the desired life-time of information in a newly generated transaction. Consequently, only recently generated transactions in the range of the window are considered to find the recently frequent item sets of a data stream.

B. Li Uncertainty pervades many application domains such as pattern recognition, sensor networks and mobile object tracking. However, in those applications, uncertain data often arrives at high speed and need to be processed in a streaming fashion. Frequent item set mining is one of the most common problems when analysing those uncertain transactions in streaming data. In this paper, we propose an efficient algorithm based on possible world semantics, called FI-UTS (Frequent Item sets mining in Uncertain Transaction Streams), for finding the set of all frequent item sets from the uncertain streaming data with a sliding window. A novel decreasing maximum count function in the algorithm is proposed to reduce the running time and the number of frequent item set to be maintained

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

when the window slides forward. Experimental results show that FI-UTS algorithm is much better than some methods for mining frequent item sets in uncertain streams.

Y. Kim, E. Park and U. Kim Frequent item set mining is a core data mining operation and has been extensively studied in a broad range of application. The frequent data stream item set mining is to find an approximate set of frequent item sets in transaction with respect to a given support threshold. In this paper, we consider the problem of approximate that frequency counts for space efficient computation over data stream sliding windows. Approximate frequent item sets mining algorithms use a user-specified error parameter, ε, to obtain an extra set of item sets that are potential to become frequent later. Hence, we developed an algorithm based on the Chern off bound for finding frequent item sets over data stream sliding window. We present an improved algorithm MAFIM (a *maximal approximate frequent item sets mining*) for frequent item sets mining based on approximate counting using previous saved maximal frequent item sets. The proposed algorithm gave a guarantee of the output quality and also a bound on the memory usage.

H. Li and S. Lee Online mining of frequent item sets over a stream sliding window is one of the most important problems in stream data mining with broad applications. It is also a difficult issue since the streaming data possess some challenging characteristics, such as unknown or unbound size, possibly a very fast arrival rate, inability to backtrack over previously arrived transactions, and a lack of system control over the order in which the data arrive. In this paper, we propose an effective bit-sequence based, one-pass algorithm, called MFI-Trans SW (*M*ining *F*requent *I*tem sets within a *Trans*action-sensitive *S*liding *W*indow), to mine the set of frequent item sets from data streams within a transaction-sensitive sliding window which consists of a fixed number of transactions. The proposed MFI-Trans SW algorithm consists of three phases: window initialization, window sliding and pattern generation. First, every item of each transaction is encoded in an effective bit-sequence representation in the window initialization phase. The proposed bit-sequence representation of item is used to reduce the time and memory needed to slide the windows in the following phases. Second, MFI-Trans SW uses the left bit-shift technique to slide the windows efficiently in the window sliding phase. Finally, the complete set of frequent item sets within the current sliding window is generated by a level-wise method in the pattern generation phase. Experimental studies show that the proposed algorithm not only attain highly accurate mining results, but also run significant faster and consume less memory than do existing algorithms for mining frequent item sets over data streams with a sliding window. Furthermore, based on the MFI-Trans SW framework, an extended single-pass algorithm, called MFI-Time SW (*M*ining *F*requent *I*tem sets within a *Time*-sensitive *S*liding *W*indow) is presented to mine the set of frequent item sets efficiently over time-sensitive sliding windows.

P.S.M. Tsai, Association rule mining is an important research topic in the data mining community. There are two difficulties occurring in mining association rules. First, the user must specify a minimum support for mining. Typically it may require tuning the value of the minimum support many times before a set of useful association rules could be obtained. However, it is not easy for the user to find an appropriate minimum support. Secondly, there are usually a lot of frequent item sets generated in the mining result. It will result in the generation of a large number of association rules, giving rise to difficulties of applications. In this paper, we consider mining top-*k* frequent closed item sets from data streams using a sliding window technique. A single pass algorithm, called *FCI_max*, is developed for the generation of top-*k* frequent closed item sets of length no more than *max_l*. Our method can efficiently resolve the mentioned two difficulties in association rule mining, which promotes the usability of the mining result in practice.

III. ALGORITHM

Here, we propose array based tail node based algorithm for mining frequent item sets using damped window model. The proposed algorithm UDS-FIM mainly consists of three procedures: (1) Damped window initialization (2) creating a global UDS-Tree; (3) mining frequent item sets from the global UDS-Tree;

A. DAMPED WINDOW INITIALIZATION

The damped window initialization phase is initiated while the number of transactions generated so far in a transaction data stream is less than or equal to a user-predefined window size w (batch). In this phase, each item of the new incoming transaction is transformed into its global UP table.

In general, after *w* batches of streaming data arrive, the proposed algorithm would traverse all O (*N*) nodes *w*−1 times and updated O(*w*2*N*) older expected support values (i.e., O(*w*2*N*) multiplications). After *w*=3 batches of streaming data arrived, the damped algorithm traversed *N*=9 nodes twice and updated 27 older expected support values (involving 27 multiplications: 9 after B2 arrived and 18 after B3 arrived).

To reduce the update cost, we propose an improved algorithm. Instead of updating older expected support values and appending the new values, our improved algorithm just appends the new expected support values to the list. Then, when computing the expected support of *X* in the data stream (of *w* batches so far, i.e., B1:*w* ≡ B1∪ ... .∪B*w*), the improved algorithm uses the following equation instead:

$$\text{expSup}(x, B_{1\ldots w}) = \sum_{j=1}^{w} (expSup(X, B_j) \times \alpha^{w-j} \qquad (1)$$

where (*X*,B*j*) is the expected support of *X* stored in the *j*-th position of the list (representing B*j* of streaming data) and $\alpha \in (0, 1]$ is the fading factor. By doing so, we avoid O(*w*2*N*) multiplications during the update process. The computation of expected support using Equation (2) involves only O(*wN*) multiplications on O(*N*) FIs.

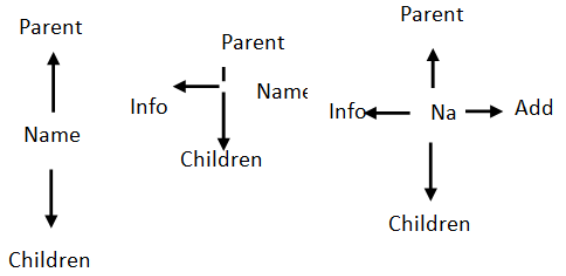
ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014

B.PARAMETERS OF UDS TREE

Let item set $X = \{x1, x2, x3\ldots xu\}$ be a sorted item set, and the item xu is called Array *tail-item* of X. When the item set X is added into a tree T in accordance with its order, the node on the tree that represents this array tail-item is called as a array *tail-node*; a node that has no children is called as a *leaf node*; a node that is neither a array tail-node nor a leaf-node is called as a *normal node.*

Before a transaction item set is added into a UDS-Tree, its corresponding probability values are appended to an



Fig(a) Structure on UDS Tree Fig(b)Tail Node tree
Fig(c)Leaf node on Global Normal Node
Figure 1. The structures of nodes on a UDS-Tree

C. ALGORITHM

Input: A Damped Tail Node Tree ***T***, a global itemsets header table ***H***, and a minimum expected support number ***minExpSN***.

Output: FIs (frequent item sets)

(1) First computing the expected support of X in the data stream (of w batches),
(2) Add the batch information on *info* field on each leaf-node to the field *add Info*;
(3) For each item x in H (from the last item) do
(4) If($x.esnT \geq$min threshold) //$x.esnT$ is from the header table H
(5) Generate an itemset $X = x$;
(6) Copy X into FIs;
(7) Create a sub header table Hx for X;
(8) If(Hx is not empty)
(9) Create a prefix UDS-Tree Tx for X;
(10) Call Sub Mining(Tx, Hx, X)
(11) Pruning non frequent item sets
(12) End if
(13) End if
(14) Pass the information of *add Info* field to parent nodes;
(15) End for
(16) Return FIs.
(17) Sub Procedure Sub Mining (Tx, Hx, X)
(18) For each item y in Hx (from the last item) do
(19) Generate an item set $Y = X \cup$;
(20) Copy Y into FIs;
(21) Create a header table Hy for Y;
(22) If (Hy is not empty)
(23) Create a prefix UDS-Tree Ty for Y;
(24) Call Sub Mining(Ty, Hy, Y)
(25) End if
(26) Pass the information of *info list* field to parent nodes;
(27) End for

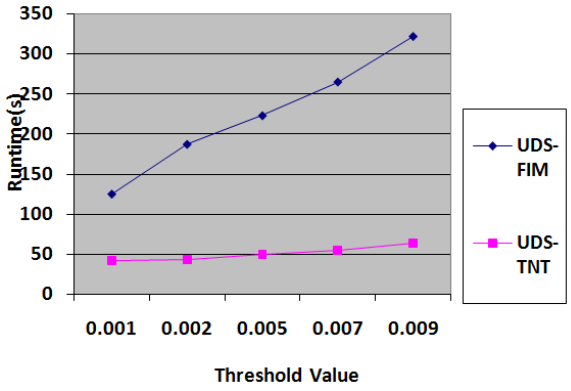
IV. PERFORMANCE EVALUATION

The proposed goal is to compare the algorithmic behavior of UDS-FIM with UDS-TNT. The performance is measured with respect to Threshold value and the number of frequent items generated .Here, We use synthetic dataset for experimental evaluation. For instance, we used an IBM synthetic data with 1M records with an average transaction length of 10 items and a domain of 1,000 items. We assigned an existential probability from the range (0,1) to every item in each transaction. We set each batch to be 10,000 transactions (for a maximum of w=200 batches). The reported figures are based on the average of multiple runs in a time-sharing environment using an 800 MHz machine. Runtime includes CPU and I/Os for mining of "frequent" patterns and maintenance of the UDS-stream structure. We evaluated different aspects of proposed algorithms, which were implemented in C.

Mining Algorithms	Minimum expected support threshold				
	0.001	0.002	0.005	0.007	0.009
UDS-FIM	453	228	143	132	88s
UDS-TNT	722	374	357	243	189

In terms of runtime, when the number of batches (w) increased, the runtime increased. Among the two algorithms, UDS-FIM took more time than the UDS-TNT because the former took slightly more time to update the expected support value due to multiplication and addition. The latter appended the expected support values of "frequent" patterns discovered from a new batch whenever the batch was processed and mined.

Synthetic T10I4D100K Dataset with Batch Size=10000
Runtime



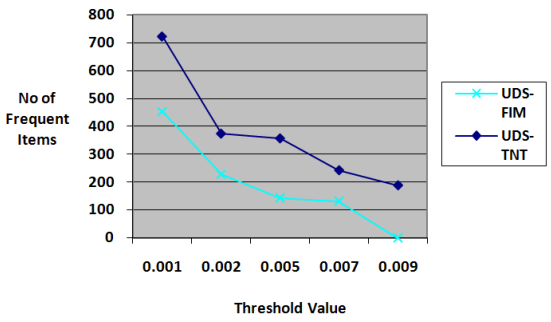
We also varied *minsup* values. When *minsup* increased, the number of expected support values stored in the UDS-FIM structure decreased because the number of "frequent" patterns mined from the stream decreased. whereas UDS TNT remains the same .

Synthetic T10I4D100K Dataset with Batch Size=10000 no of frequent item

Mining Algorithms	Minimum expected support threshold				
	0.001	0.002	0.005	0.007	0.009
UDS-FIM	125.146	187.37	223.23	265.42	322.43
UDS-TNT	42.181	43.479	49.941	54.43	63.37

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 9, September 2014



V. CONCLUSION

In this paper, we proposed tree-based mining algorithms UDS –FIM that can be used for mining frequent patterns from dynamic streams of uncertain data and compared with UDS-TNT. The UDS-FIM maintains the frequent items in the UDS tree with pre min sup to find frequent item sets. The mined items are then stored in the stream structure with the expected support values. When the next batch arrives, it updates the UDS-stream structure. The UDS-FIM works for various min sup value whereas UDS-TNT does not.

REFERENCES

[1] K.Kannika Parameswari and Dr.Antony Selvadoss Thanamami,"A new Algorithm for Finding FIs Using Damped Window Model",IJIRCCE,Vol 2,No 8,pg No.5515-5520, October 2013.

[2] K.Jothimani and Dr.Antony Selvadoss Thanamami,"EDS-FI:Efficient Data Structure for Mining Frequent Itemsets",under Informationand Communication Technology including computer Science(ICT)from ISCA(Indian Science Congress Association),Kolkatta 3rd -7th January,2013.

[3] K.Jothimani and Dr.Antony Selvadoss Thanamami ,"Determining the factors for mining Frequent Itemsets in Data SDtreams",NCCICT held at VEL TECH Dr.RR & SR Technical University,Avadi during 12th -13th August,2011,pp-127-130,ISBN 978-93-80624-43-3.

[4] Aggarwal, C.C., Li, Y.,Wang, J.,Wang, J.: Frequent pattern mining with uncertain data. In: ACM KDD, pp. 29–37 (2009)

[5] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB, pp. 487–499. Morgan Kaufmann, San Francisco (1994)

[6] Calders, T., Garboni, C., Goethals, B.: Efficient pattern mining of uncertain data with sampling. In: Zaki, M.J., Yu, J.X., Ravindran, B., Pudi, V. (eds.) PAKDD 2010, Part I. LNCS (LNAI), vol. 6118, pp. 480–487. Springer, Heidelberg (2010)

[7] C.C. Aggarwal and P.S. Yu, "A survey of uncertain data algorithms and applications," IEEE Transactions on Knowledge and Data Engineering, Vol.21, no.5, pp.609-623, 2009.

[8] C.K. Leung, M.A.F. Mateo and D.A. Brajczuk, A tree-based approach for frequent pattern mining from uncertain data, in 12th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2008). 2008, pp.653-661.

[9] K.Jothimani and Dr.Antony Selvadoss Thanamami," CB Based Approach for Mining Frequent Itemsets", International Journal of Modern Engineering Research (IJMER), ISSN: 2249-6645, Vol.2, Issue.4, July-Aug. 2012 pp-2508-2511.

[10] X. Sun, L. Lim and S. Wang, "An approximation algorithm of mining frequent itemsets from uncertain dataset," International Journal of Advancements in Computing Technology, Vol.4, no.3, pp.42-49, 2012.

[11] R. Agrawal, T. Imielinski and A. Swami. "Mining association rules between sets of items in large databases". In Proc. of the ACM SIGMOD Conference on Management of Data, Washington, D.C, May 1993

[12] L. Wang, D.W. Cheung, R. Cheng, S. Lee, and X. Yang, "Efficient Mining of Frequent Itemsets on Large Uncertain Databases," IEEE Transactions on Knowledge and Data Engineering, no.99(PrePrints), 2011.

[13] C.K. Leung, C.L. Carmichael and B. Hao, Efficient mining of frequent patterns from uncertain data, in International Conference on Data Mining Workshops (ICDM Workshops 2007). 2007, pp.489-494.

[14] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD, pp. 1–12 (2000)

[15] C. K.-S. Leung, M. A. F. Mateo, and D. A. Brajczuk. A tree-based approach for frequent pattern mining from uncertain data. In PAKDD, pages 653–661, 2008.

[16] Jiang, N., Gruenwald, L.: Research issues in data stream association rule mining. SIGMOD Record 35(1), 14–19 (2006)

[17] Leung, C.K.-S.: Mining uncertain data. WIREs Data Mining and Knowledge Discover 1(4), 316–329. John Wiley & Sons, Hoboken, NJ (2011)

BIOGRAPHIES

K.Kannika Parameswari is a research scholar in Department of Computer Science, Nallamuthu Gounder Mahalingam College, Pollachi. She received her Master of Computer Applications (MCA) from Karpagam College of Engineering, Affiliated to Anna University, Coimbatore. she worked as an Assistant Professor in the Department of MCA in Karpagam College Of Engineering ,Coimbatore. She has presented paper in National Conference and attended Workshop/Seminars. Her research focuses on Data Mining.

Dr. Antony Selvadoss Thanamani is presently working as Professor and Head, Dept of Computer Science, NGM College, Coimbatore, India (affiliated to Bharathiar University, Coimbatore). He has published more than 100 papers in international/ national journals and conferences. He has authored many books on recent trends in Information Technology. His areas of interest include E-Learning, Knowledge Management, Data Mining, Networking, Parallel and Distributed Computing. He has to his credit 24 years of teaching and research experience. He is a senior member of International Association of Computer Science and Information Technology, Singapore and Active member of Computer Science Society of India, Computer Science Teachers Association, New York.